IN THE SPECIFICATION:

Please amend the paragraph beginning at page 3, line 2, as follows.

Generally, a method and apparatus are disclosed for locking the most recently accessed frames in a cache memory. The most recently accessed frames in a cache memory are likely to be accessed by a task again in the near future. Thus, the most recently used frames may be locked in accordance with the present invention at the beginning of a task switch or interrupt, either with a special instruction or by automatic means, to improve the performance of the cache. The list of most recently used frames is updated as a task executes. The list of most recently used frames may be embodied, for example, as a list of ~~frames~~ frame addresses or a flag associated with each frame. The list of most recently used frames may be separately maintained for each task if multiple tasks may interrupt each other.

Please amend the paragraph beginning at page 4, line 2, as follows.

FIG. 1 illustrates an adaptive frame locking mechanism 100 in accordance with the present invention. As shown in FIG. 1, the adaptive frame locking mechanism 100 selectively locks frames of a cache 150, such as frame 1 in set 0, frame 2 in sets 2 and n-1 and frame N in set N. According to one aspect of the present invention, frames are identified for locking that, at any particular time, are likely to be accessed by a task in the near future. Once locked, the information contained in a frame cannot by displaced by an interrupting task. According to a further aspect of the present invention, the ~~fame~~ frame locking mechanism adapts to the frame use of a particular task. In addition, as discussed further below, the present invention provides a mechanism that automatically unlocks frames that cause significant performance degradation for a running task.

Please amend the paragraph beginning at page 5, line 28, as follows.

A write back cache may contain data written into it that has not yet been written into main memory. A block in a cache that is more recent than its image in main memory is termed "dirty." The dirty block is written to main memory when it is evicted, or if another process ~~on~~ or another processor accesses the block in a coherent memory scheme. If another process attempts to gain ownership of a dirty block, the dirty block is

written back to memory and then its frame is invalidated in the cache. If another process attempts to read data in a dirty block, the dirty block is written back to memory, and thereby becomes "clean," but the frame containing the block is not invalidated. These actions occur whether the frame is locked or not. Therefore, no special action is necessary in a write back data cache if dirty blocks are in locked frames. The only additional action that must occur in a cache that uses a frame locking scheme relative to one that does not use a frame locking scheme, is that a locked frame that is invalidated is unlocked, as would be expected, to free it for subsequent use by any task.

Please amend the paragraph beginning at page 6, line 18, as follows.

As shown in FIG. 3, the exemplary most recently used frame circuit 300 includes three one-bit latches 310-i, 320-i, 330-i (hereinafter, referred to as latches "a," "b," and "locked") ~~are~~ associated with each frame, i. Latch a is set when the frame that it is associated with is accessed. After every n accesses, the value in latch a of a frame is transferred to latch b and latch a is reset. Therefore, if a frame has been accessed during the last n accesses, latch b will be set. Following a transfer, latch a is again set when its frame is accessed. This operation continues during the execution of a program.